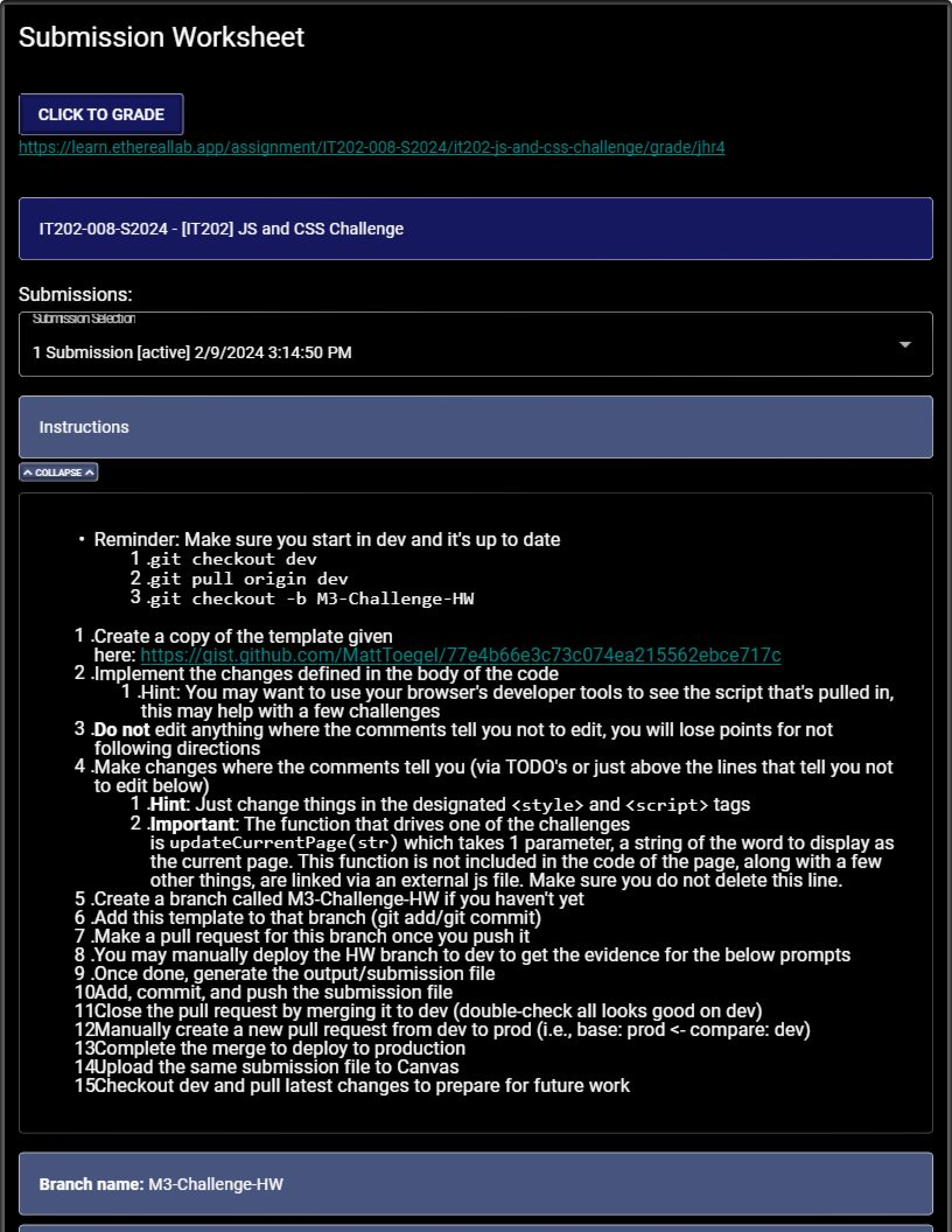


Submission Worksheet

https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-js-and-css-challenge/grade/jhr4

IT202-008-S2024 - [IT202] JS and CSS Challenge

Submissions:

Submission Selection

1 Submission [active] 2/9/2024 3:14:50 PM

Instructions

- Reminder: Make sure you start in dev and it's up to date
    1. `git checkout dev`
    2. `git pull origin dev`
    3. `git checkout -b M3-Challenge-HW`

1. Create a copy of the template given here: https://gist.github.com/MattToegel/77e4b66e3c73c074ea215562ebce717c
2. Implement the changes defined in the body of the code
    1. Hint: You may want to use your browser's developer tools to see the script that's pulled in, this may help with a few challenges
3. **Do not** edit anything where the comments tell you not to edit, you will lose points for not following directions
4. Make changes where the comments tell you (via TODO's or just above the lines that tell you not to edit below)
    1. **Hint**: Just change things in the designated `<style>` and `<script>` tags
    2. **Important**: The function that drives one of the challenges is `updateCurrentPage(str)` which takes 1 parameter, a string of the word to display as the current page. This function is not included in the code of the page, along with a few other things, are linked via an external js file. Make sure you do not delete this line.
5. Create a branch called M3-Challenge-HW if you haven't yet
6. Add this template to that branch (git add/git commit)
7. Make a pull request for this branch once you push it
8. You may manually deploy the HW branch to dev to get the evidence for the below prompts
9. Once done, generate the output/submission file
10. Add, commit, and push the submission file
11. Close the pull request by merging it to dev (double-check all looks good on dev)
12. Manually create a new pull request from dev to prod (i.e., base: prod <- compare: dev)
13. Complete the merge to deploy to production
14. Upload the same submission file to Canvas
15. Checkout dev and pull latest changes to prepare for future work

Branch name: M3-Challenge-HW

Tasks: 15 Points: 10.00

● Screenshots (4 pts.)

^ COLLAPSE ^

●

^ COLLAPSE ^

Task #1 - Points: 1

Text: 5 Screenshots based on checklist items

Checklist *The checkboxes are for your own tracking

#	Points	Details
☑ #1	1	A screenshot showing the Primary page with the checklist items completed (the view that initially loads)
☑ #2	1	A screenshot showing the page after the login link is clicked with URL shown
☑ #3	1	A screenshot showing the page after the register link is clicked with URL shown
☑ #4	1	A screenshot showing the page after the profile link is clicked with URL shown
☐ #5	1	A screenshot showing the page after the logout link is clicked with URL shown
☑ #6	1	Screenshots should be from heroku dev

Task Screenshots:

Large Gallery

Checklist Items (2)

#1 A screenshot showing the Primary page with the checklist items completed (the view that initially loads)

#6 Screenshots should be from heroku dev

Screenshot of Primary Page with Heroku Dev link on top.

Checklist Items (2)

#2 A screenshot showing the page after the login link is clicked with URL shown

#6 Screenshots should be from heroku dev

Screenshot of Page after Login is clicked with Heroku Dev link on top.

Checklist Items (2)

#3 A screenshot showing the page after the register link is clicked with URL shown

#6 Screenshots should be from

Checklist Items (2)

#4 A screenshot showing the page after the profile link is clicked with URL shown

#6 Screenshots should be from

heroku dev

heroku dev

Screenshot of Page after Register is clicked, with Heroku Dev link on top.

Screenshot of Page after Profile link clicked, with Heroku Dev link on top.

Checklist Items (2)

#5 A screenshot showing the page after the logout link is clicked with URL shown

#6 Screenshots should be from heroku dev

Screenshot of Page after Logout is clicked, with Heroku Dev link on top.

● Explanations (4 pts.)

^ COLLAPSE ^

● ^ COLLAPSE ^

Task #1 - Points: 1

Text: Briefly explain how you made the navigation horizontal

Details:

Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

To create the horizontal navigation, I targeted the list in nav by "nav li {}" and changed the display style to inline which over wrote the default value of lists (probably block) and made it inline which made it go in a line horizontally.

● ^ COLLAPSE ^

Task #2 - Points: 1

Text: Briefly explain how you remove the navigation list item markers

Details:

Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I removed the navigation list item markers by making making the list-style-type to none. This property is used to change the bullet point to something else and none removes it. I targeted only the nav bullet points by nav ul {} which

change the bullet point to something else and none removes it. I targeted only the nav bullet points by nav ul {} which was the unordered list in nav.

Task #3 - Points: 1

Text: Briefly explain how you gave the navigation a background

ⓘDetails:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I gave the navigation a background by using the background-color: bisque; under the nav{} which meant only the nav got the background color of bisque.

Task #4 - Points: 1

Text: Briefly explain how you made the links (or their surrounding area) change color on mouseover/hover

ⓘDetails:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I used the pseudo class of :hover for the link tag <a> and made the text color change by "color:lightcoral". I targeted the links in nav by just nav a:hover {} I could of just done a:hover too as there's no other links on the page.

Task #5 - Points: 1

Text: Briefly explain how you changed the challenge list bullet points to checkmarks (✓)

ⓘDetails:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I changed the challenge list bullet points to checkmark's by using the same property of "list-style-type:" but instead of

using none I used the unicode number for the check mark which was 2713 so the property was "list-style-type: "\2713";"
To target the bullet points only in the directions, I added targeted the all the ul {} however, I placed it before the step #2 in which I removed the item markers. Therefore, when the code ran it changed all the unordered list bullet points to checkmarks and then changed the bullet points in nav to nothing. Therefore, in theory just the bullet points in the directions were impacted at the end.

Task #6 - Points: 1

Text: Briefly explain how you made the first character of the h1 tags and anchor tags uppercased

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I made the first character of the h1 tags and anchor tags uppercased by using text-transform which is used to transform the text in the selected elements to many different things. I set the text-transform value to capitalize because this value makes the first character capitalized. As I had to target all the h1 and anchor tags I just used h1, a { }which selected all the <h1> tags and <a>.

Task #7 - Points: 1

Text: Briefly explain/describe your custom styling of your choice

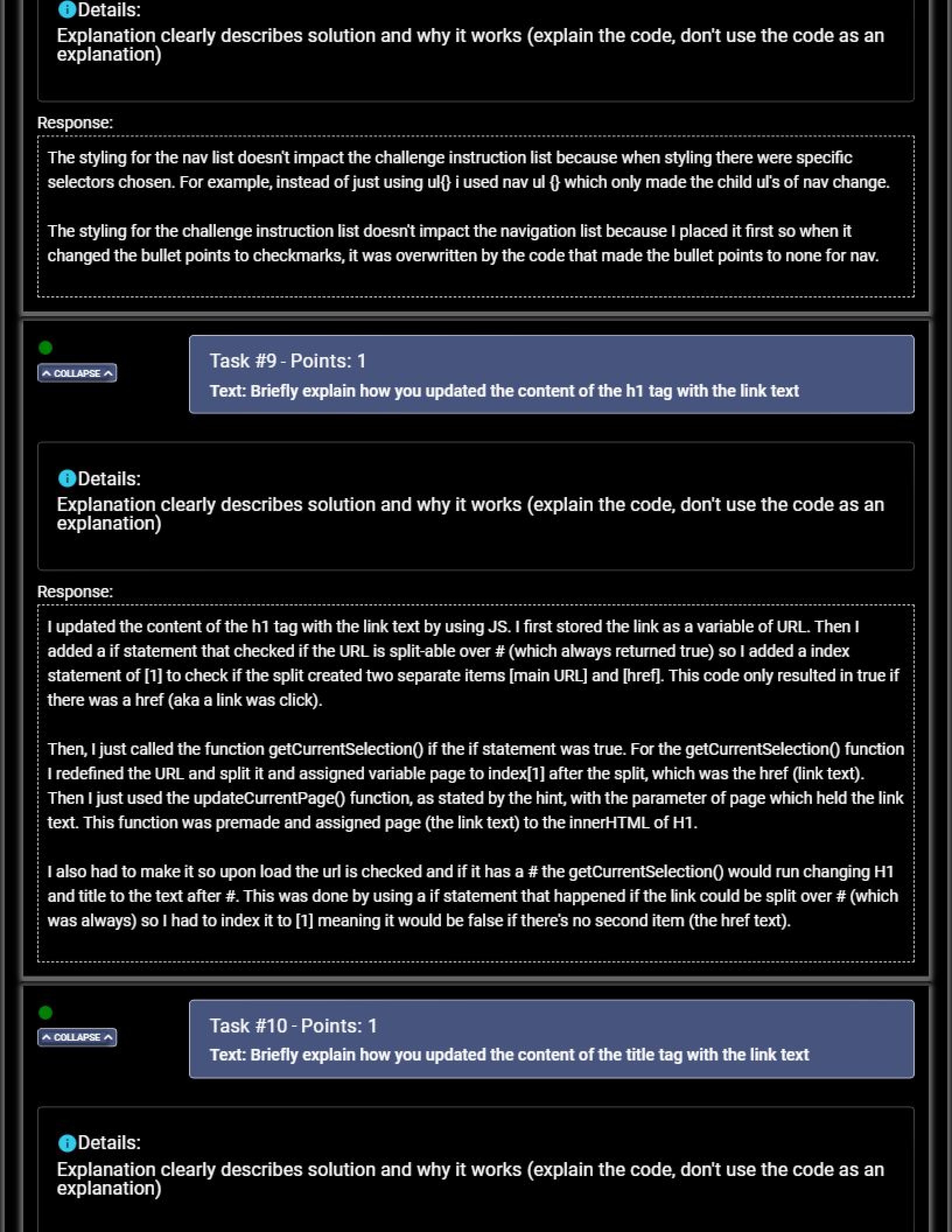
Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

For the custom styling I just added a border to the nav by using the border property and set the values to 1em (width/weight) solid (outline) and burlywood (color). I also made the text larger by using the font-size property which is used to change the size. I set it to xx-large.

Task #8 - Points: 1

Text: Briefly explain how the styling for the challenge list doesn't impact the navigation list



Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

The styling for the nav list doesn't impact the challenge instruction list because when styling there were specific selectors chosen. For example, instead of just using ul{} i used nav ul {} which only made the child ul's of nav change.

The styling for the challenge instruction list doesn't impact the navigation list because I placed it first so when it changed the bullet points to checkmarks, it was overwritten by the code that made the bullet points to none for nav.

Task #9 - Points: 1

Text: Briefly explain how you updated the content of the h1 tag with the link text

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I updated the content of the h1 tag with the link text by using JS. I first stored the link as a variable of URL. Then I added a if statement that checked if the URL is split-able over # (which always returned true) so I added a index statement of [1] to check if the split created two separate items [main URL] and [href]. This code only resulted in true if there was a href (aka a link was click).

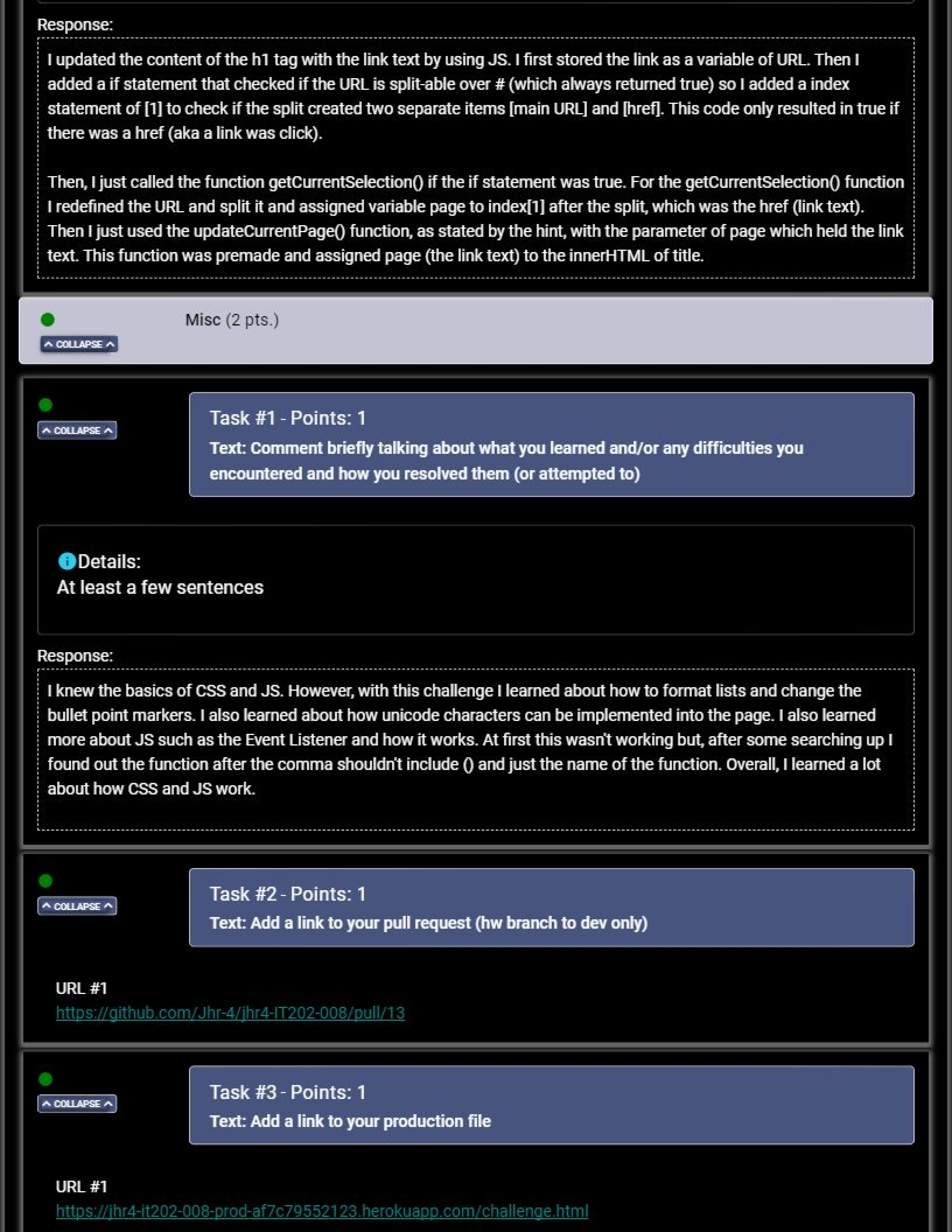
Then, I just called the function getCurrentSelection() if the if statement was true. For the getCurrentSelection() function I redefined the URL and split it and assigned variable page to index[1] after the split, which was the href (link text). Then I just used the updateCurrentPage() function, as stated by the hint, with the parameter of page which held the link text. This function was premade and assigned page (the link text) to the innerHTML of H1.

I also had to make it so upon load the url is checked and if it has a # the getCurrentSelection() would run changing H1 and title to the text after #. This was done by using a if statement that happened if the link could be split over # (which was always) so I had to index it to [1] meaning it would be false if there's no second item (the href text).

Task #10 - Points: 1

Text: Briefly explain how you updated the content of the title tag with the link text

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)



Response:

I updated the content of the h1 tag with the link text by using JS. I first stored the link as a variable of URL. Then I added a if statement that checked if the URL is split-able over # (which always returned true) so I added a index statement of [1] to check if the split created two separate items [main URL] and [href]. This code only resulted in true if there was a href (aka a link was click).

Then, I just called the function getCurrentSelection() if the if statement was true. For the getCurrentSelection() function I redefined the URL and split it and assigned variable page to index[1] after the split, which was the href (link text). Then I just used the updateCurrentPage() function, as stated by the hint, with the parameter of page which held the link text. This function was premade and assigned page (the link text) to the innerHTML of title.

● Misc (2 pts.)

Task #1 - Points: 1

Text: Comment briefly talking about what you learned and/or any difficulties you encountered and how you resolved them (or attempted to)

Details:
At least a few sentences

Response:

I knew the basics of CSS and JS. However, with this challenge I learned about how to format lists and change the bullet point markers. I also learned about how unicode characters can be implemented into the page. I also learned more about JS such as the Event Listener and how it works. At first this wasn't working but, after some searching up I found out the function after the comma shouldn't include () and just the name of the function. Overall, I learned a lot about how CSS and JS work.

Task #2 - Points: 1

Text: Add a link to your pull request (hw branch to dev only)

URL #1
https://github.com/Jhr-4/jhr4-IT202-008/pull/13

Task #3 - Points: 1

Text: Add a link to your production file

URL #1
https://jhr4-it202-008-prod-af7c79552123.herokuapp.com/challenge.html

● ^ COLLAPSE ^

Task #4 - Points: 1

Text: Waka Time (or related) Screenshot

Checklist *The checkboxes are for your own tracking

#	Points	Details
■ #1	1	Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated with the grade for this item)

Task Screenshots:

○ Large Gallery

Checklist Items (1)

#1 Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated with the grade for this item)

It shows how challenge html was worked on which is in the public_html folder inside of the M3-Challenge-HW branch.